DirectDSP Software

Users Guide for Linux

Copyright ã Signalogic, Inc.
Revision C
April, 2003

Signalogic, Inc.
9617 Wendell @ Skillman
Dallas, TX 75243
Tel: 214-349-5551
Fax: 214-343-0163

http://www.signalogic.com
tech_support@signalogic.com

TABLE OF CONTENTS

1 System Overview 3
1.1 Hardware 3
1.2 Software 3
1.3 Manual Notes and Conventions 3
1.4 Hardware Installation and Checkout 4
1.5 Directory Structure 7
1.6 Linux Driver Install and Run Instructions 8
2 Demo Programs 9
2.1 Running the E1-Audio Demo Program 9
2.2 Running the DUART Demo Program 15
3 Rebuilding Software 17
3.1 Rebuilding the E1-Audio Demo Program 17
3.2 Rebuilding the DUART Demo Program 18
3.3 Rebuilding the LKM Driver 18
3.4 Rebuilding Engine Manager 19
3.5 Rebuilding Hardware Library 19

1 System Overview

This document assumes you are using a Pentium-compatible system to run the Signalogic SigC54xx and SigC67xx PC/104 board demo programs for I/O peripherals and other board resources. The Pentium-compatible system should be configured with:

1.1 Hardware

- Pentium-compatible main PC/104 board, for example standard desktop system using ISA passive carrier card, or embedded PC/104 system such as Lippert CoolRunner II or WinSystems
- SigC54xx-PC/104 board or SigC67xx-PC/104, Rev. C1
- SigC5416-SODIMM module, Rev. B, or SigC6711-SODIMM module, Rev. B
- SigSD4 Audio Module, Rev. C (optional, depending on application requirements)
- Optional digital I/O board or other third-party boards

1.2 Software

- Red Hat Linux v7.3, Kernel v2.4.18-3
- Signalogic LKM driver, v1.2
- Signalogic shared object module Hardware Library, v1.2
- Signalogic shared object module Engine Manager, v1.2
- Signalogic shared object module Hardware Manager, v1.2

1.3 Manual Notes and Conventions

Command line examples in the following sections beginning with

```
shell/>
```

indicate a command that needs to be typed in at the shell command prompt.

1.4 Hardware Installation and Checkout

The SigC54xx-PC/104 board is assumed to be at the top of the PC/104 board stack, as shown below.

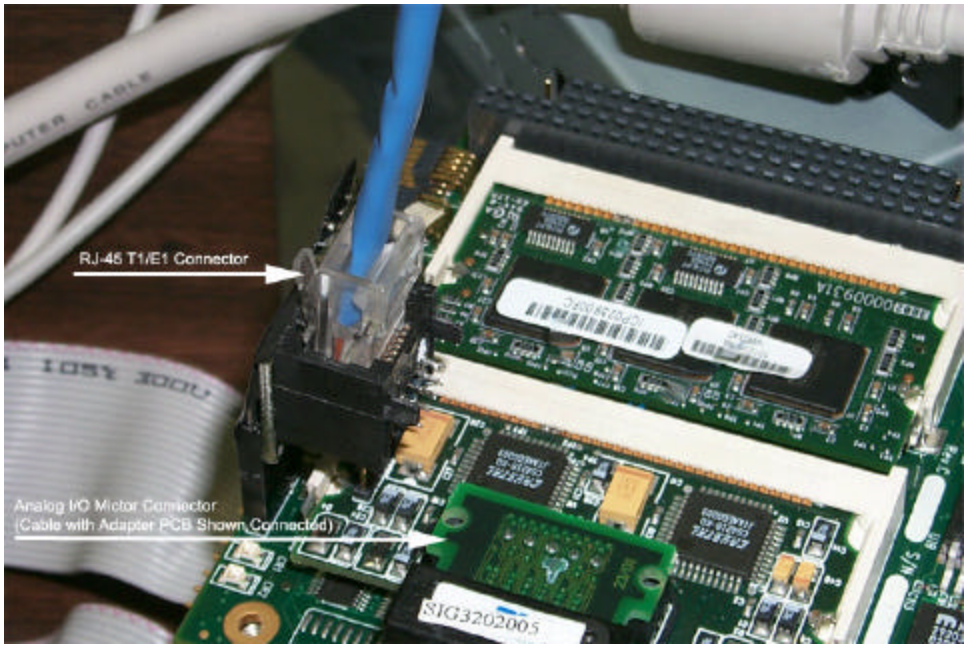

Figure 1 SigC54xx-PC/104 board stack installation

In the above picture, note the position the RJ-45 / magnetics adapter PCB, mounted at left side of the SigC54xx-PC/104 board. The E1 loopback cable and plug is shown inserted (blue cable). The plug cable is about 3" long with wires exposed at one end and has RXRING tied to TXTIP and RXTIP tied to RXRING.

Note also the position of the audio I/O Mictor cable connector, shown connected to the SigSD4 audio module (2nd module socket, at bottom).

Below are installation items that should be verified prior to running the system.

1. For some Pentium PC/104 main boards, the hard drive (HD) connector can be very unstable. For example, on a Lippert CoolRunner II board, the HD cable will come loose at the slightest twitch. ***Watch out for this!!*** The cable should be securely attached and pushed down fully onto the main board.

2. The SigC54xx-PC/104 board base I/O address default value is 0x320. In some cases, this value may need to be changed to avoid conflict with another board in the stack. For example, in the Lippert system the SigC54xx-PC/104 board base address must be set to 0x340 to avoid conflict with the RS-232 digital I/O board.

To change or verify the SigC54xx board base I/O address, check the rotary hex switch on the

board to make sure it is set to the middle digit of the base I/O address. For example, for base I/O address 0x320 the switch should be set to “2”, and for base I/O address 0x340 the switch should be set to "4".

3. Note that the video on the Lippert system main board has been set to 800 x 600 x 4 due to the low amount of memory (less than 2 Mbyte) reserved for video display. This may make some GUI dialogs and screen displays look "nearly monochrome", but it's the best that can be done given system limitations.

4. For E1-audio loopback testing, be sure that the loopback plug is securely inserted into the RJ-45 adapter PCB. ***The loopback plug should be pushed all the way in*** and not allowed to rock back and forth.

5. For DUART testing, be sure that the RS-232 adapter is securely inserted into the DB9 adapter PCB header at J35 (topmost RS-232 header on the right side of the board). J35 contains signal for DUART channel A; J34 contains signal for DUART channel B. For lab testing purposes, use a DB9 female-to-female serial cable to attach the RS-232 adapter to another PC’s COM port. See Figures 2 and 3 below for additional explanation.

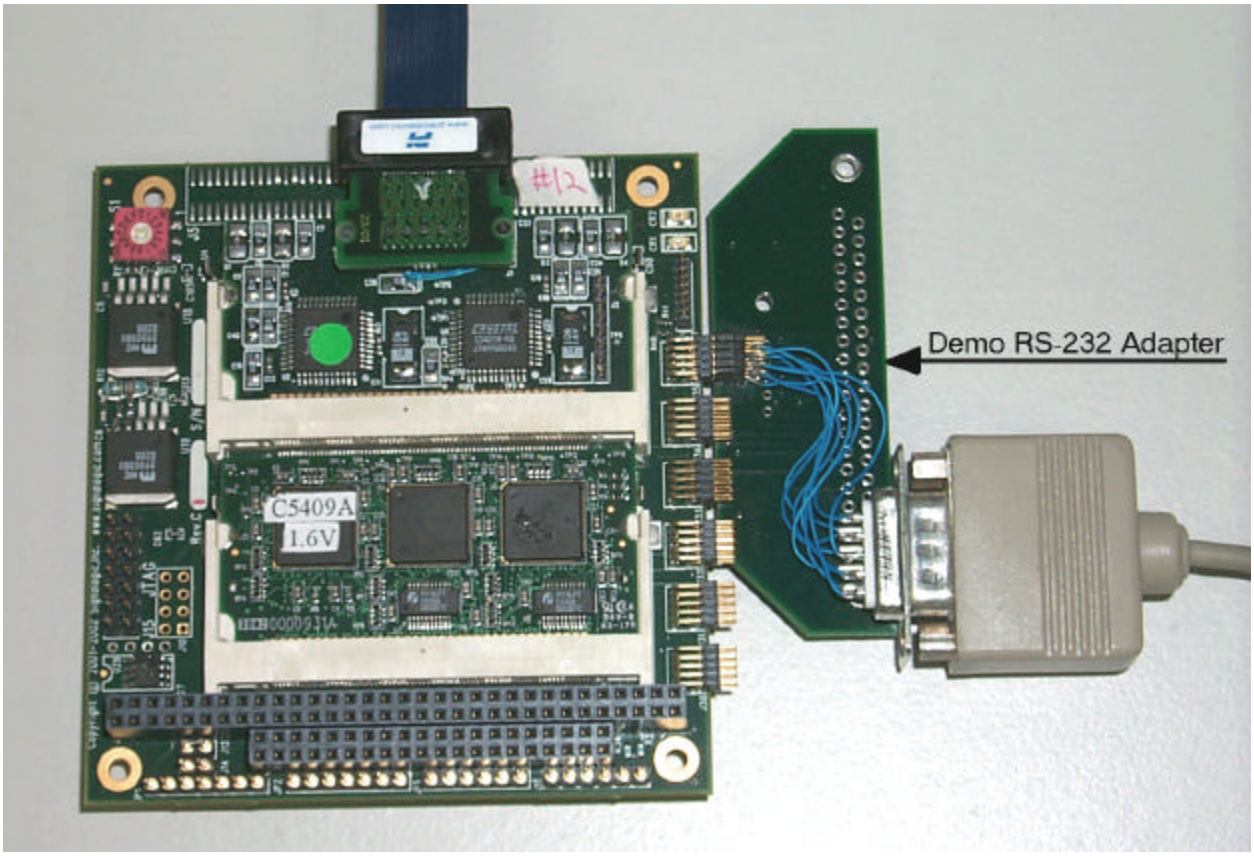

Figure 2 SigC54xx-PC/104 with DUART adapter shown

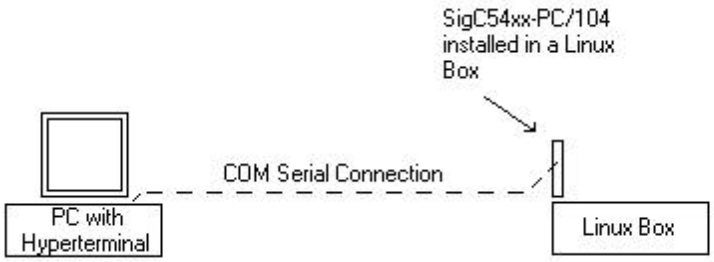

Figure 3 SigC54xx-PC/104 with COM Serial Connection to Hyperterminal PC or other TTY Terminal Emulator

1.5 Directory Structure

There should be three directories under the main extraction directory, as listed below:

1. SigFModule
2. EngMgr
3. HwMgr

By default, the main extraction directory is:

```
/usr/local/share
```

Note that the SigFModule folder contains files for both Hardware Library shared object module and the LKM driver.

1.6 Linux Driver Install and Run Instructions

To install the SigC54xx / SigC67xx board loadable kernel module (LKM) driver, follow the steps below:

1) Log-in as root:

```
Login: root
Password: veyung
```

2) Install the loadable kernel module:

```
shell/> cd /usr/local/share/SigFModule/bin
shell/> insmod SigFM.o
```

3) Change back to the main extraction directory by entering:

```
shell/> cd ../..
```

To rebuild the loadable kernel module driver, follow the instructions in section 3.3 below.

NOTE: *currently the LKM driver must be installed each time the system is rebooted. However, in the future Signalogic engineers plan to make the driver load automatically, by editing the /etc/modules.conf file and adding "module probing" of the SigC54xx board at boot time.*

2 Demo Programs

A series of demo programs can be used to test the SigC54xx and SigC67xx PC/104 boards. These programs also can be used for diagnostic purposes for individual processor and I/O peripheral resources on the boards.

2.1 Running the E1-Audio Demo Program

***NOTE**: The following instructions assume that the SigFM.o module has been built from the SigFModule directory, and that libEngMgr.so has been built from the EngMgr directory. If not, see build instructions in section 3.3.*

After initial system reboot, the kernel module must be loaded. See instructions in section 1.6 above, Linux Driver Install and Run Instructions.

To run the E1-Audio demo program, first enter:

```
shell/> startx
```

This will start X Windows, a Linux graphical user interface. After X Windows starts up, holding down the left mouse button (or right, depending on which window manager program is running) will bring up a menu. Choose "XTerm" from the menu. This will bring up a small window that represents a console. The following commands must be typed in from this console window:

```
shell/> /sbin/insmod SigFM
```

This will install the SigFM.o module into the kernel so that the Signalogic software can run. Note that XTerm is an option only under X Windows, and not KDE, which is the default window manager provided with Red Hat Linux. On systems with relatively small amount of memory, such as the Lippert CoolRunner II, KDE may not be used due to memory considerations.

Then move to the subdirectory from which the E1-Audio demo program will be run, as follows:

```
shell/> cd /usr/local/share/HwMgr
```

To run the Signalogic E1-Audio demo program, type in the XTerm window:

```
shell/Hwmgr> ./e1test
```

This will run the E1-Audio version of XDScope, one of the demo programs provided with the Linux version of DirectDSP software. If e1test.o runs successfully, the Hardware Manager dialog box will appear as shown below:

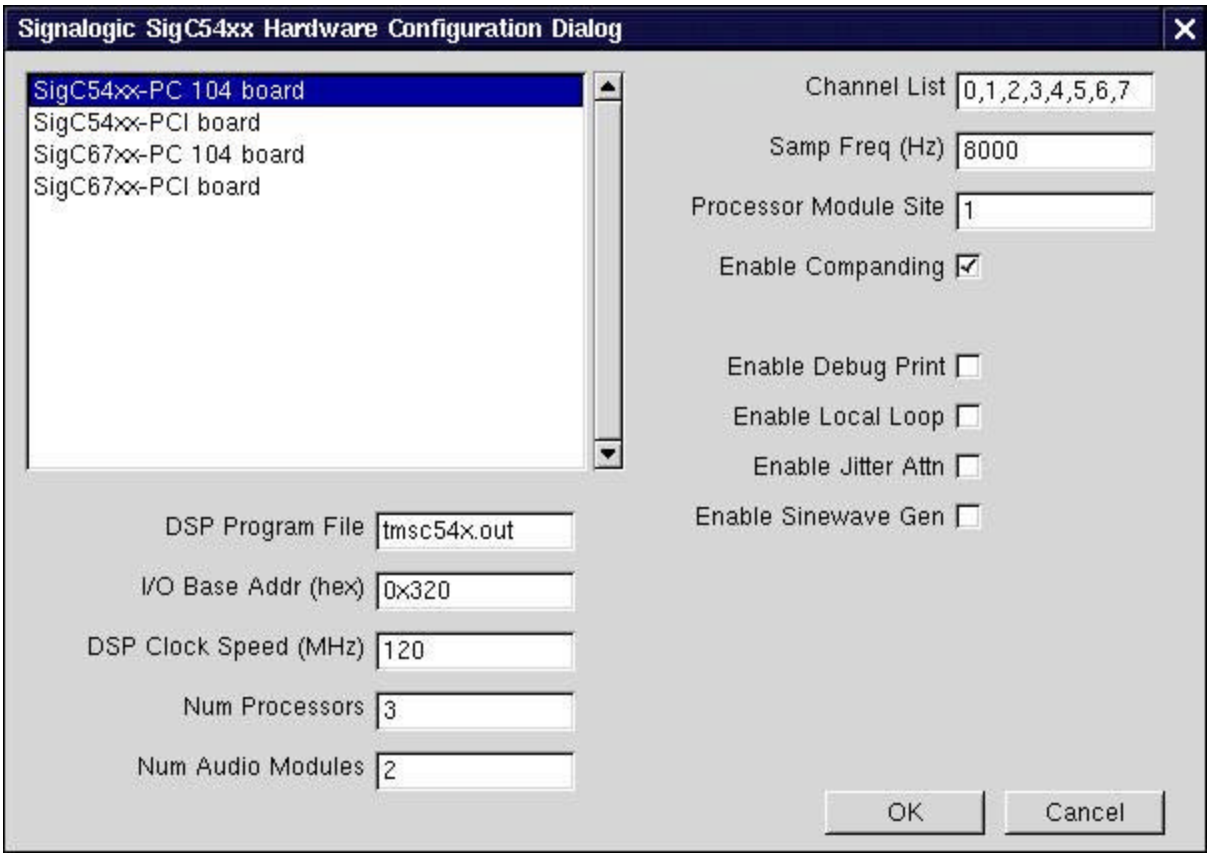

Figure 4 Hardware Manager dialog

Note that in Figure 4 above, the XTerm window is also visible, showing the command-line entry used to invoke the E1-Audio demo program (e1test.o). In Figure 7, below, more of the background screen is shown, with displays visible from XTerm and other steps preceding Hardware Manager dialog display.

Below are some notes about particular fields in the Hardware Manager dialog box:

- the DSP program file should be set to its default value of tmsc54x.out
- the default base I/O address for the SigC54xx-PC/104 board is 0x320, but should be set to 0x340 when installed in the Lippert system
- the Processor Clock Speed field should be set to:

  100 MHz for C5409 and C5402 modules

  120 MHz for C5409A module

  160 MHz for C5416 modules

  where the module type refers to DSP module installed on the SigC54xx-PC/104 board.

Note that Processor Clock Speed field values should be entered in MHz, for example:

```
100
120
160
```

- the Number of Processors field should correspond to the number of DSP devices installed on the DSP module. The valid range is 1 to 3. If you are not sure, enter 1 which will cause only the first DSP to be activated.
- the Sampling Rate field should be set to 8000 Hz for T1/E1 use (for non T1/E1 use, valid sampling rate range is 4000 to 48000 Hz)
- the Channel List field should contain the desired channels to be displayed, in order of onscreen trace display from top to bottom, with each channel being separated by a comma. The allowable range for channels is 0-7 and the maximum number of traces is 8 (eight). Some example Channel List field entries follow:

```
0,1,2,3           (first four channels displayed)

0,0,3             (channel 0 displayed on traces 0 and 1,
                   channel 3 displayed on trace 2)

1                 (channel 1 displayed only)
```

- for E1-audio testing, the "Enable Companding" checkbox should be set. This field will cause the DSP to enable A-law companding on McBSP1 data, which is normally routed to the T1/E1 device on the SigC54xx-PC/104 board. On the transmit side, data is A-law compressed and bit inverted; on the receive side, Rx data is bit-inverted and A-law expanded.
- to run the E1-Audio demo program and cause debug information error and status messages to also appear in the XTerm command window (in addition to standard audio trace display), set the Enable Debug checkbox

  NOTE: *if you check Enable Debug, then you audio trace display may be slow because of the additional information being read from the SigC54xx DSP module and displayed onscreen (see Figure 6 below).*

Click "OK" at lower right of Hardware Manager to begin the E1-Audio demo program. You should see a digital scope style display as shown below, with XDScope shown in the window title bar. You may need to wait from 1 to 5 seconds for display traces to become active, depending on the speed of your system.

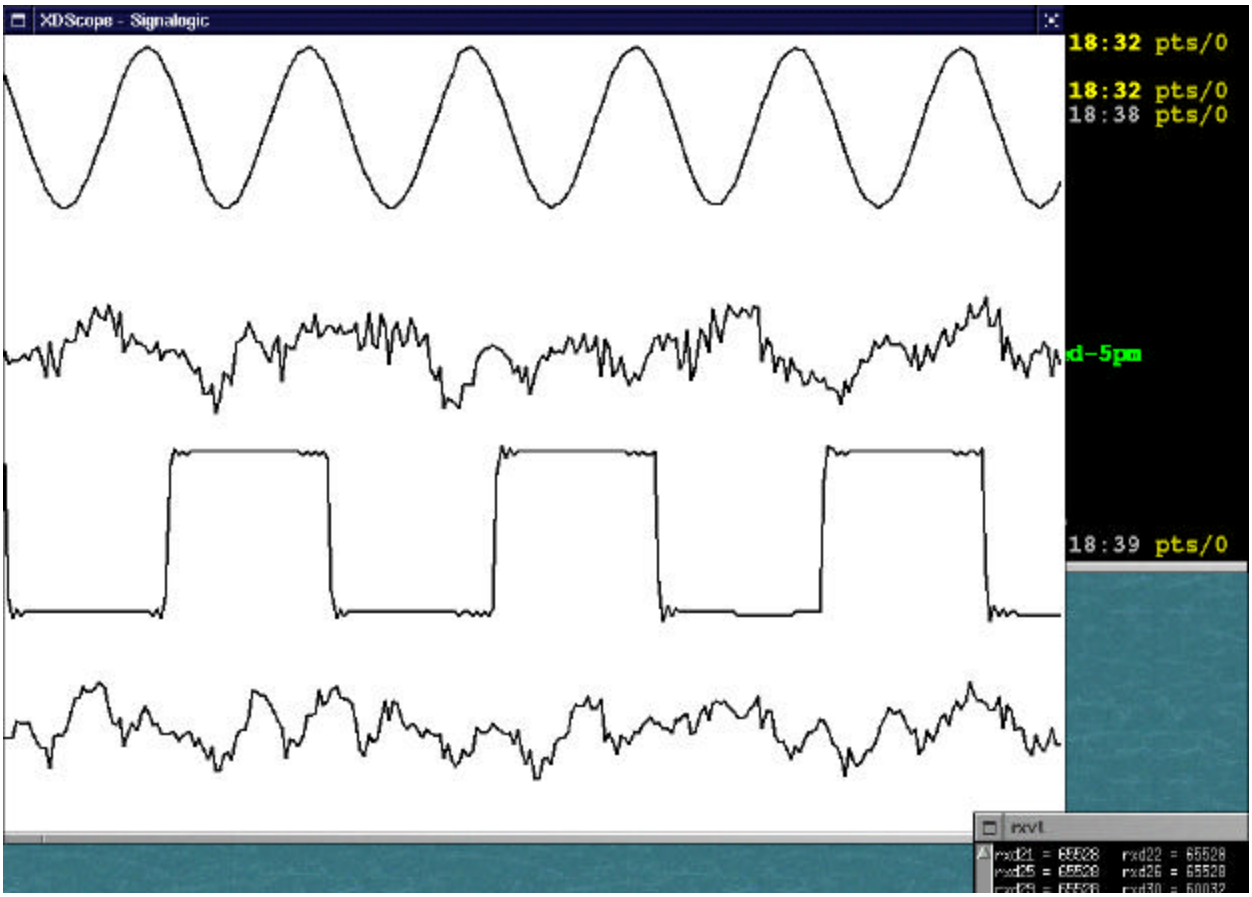

Figure 5 XDScope display of E1-audio data

In the above display, test pattern data is shown on traces 1 and 3, and speech data is shown on traces 2 and 4. Up to N data traces received from E1 line input are displayed simultaneously by XDScope software, where N is the number of channels entered in the Channel List field of the Hardware Manager dialog box (Figure 5 above).

Note that for sine wave test data, it may be possible to see the results of A-law companding over E1 channels, in the form of a “stair step” effect at higher signal amplitudes.

In E1 wire-loopback mode (with E1 loopback plug installed in RJ-45 connector), each trace corresponds to the appropriate input channel from the SigSD4 Audio Module. For example, audio input data on SigSD4 input channel 0 will appear (after being transmitted and received over E1) on the first trace, channel 1 on the second trace, etc.

The demo can be exited at any time by pressing "Esc", with the display window in the foreground (i.e. the active window).

The E1-Audio demo program performs the following functions:

- initializes the SigC54xx board, C54xx DSP processor, and memory-map

- initializes the SigSD4 audio module
- initializes the Dallas Semiconductor DS2155 T1/E1 device
- downloads executable DSP code (COFF file format)
- sets DSP properties that control setup of onboard resources
- runs the DSP, begins audio and E1 processing
- displays continuous buffers of analog input data sent through the E1 circuit
- outputs audio data shown on the display traces

Below is another XDScope screen capture, showing a debug information print out window. To enable debug information display, set the Enable Debug Info checkbox in the Hardware Manager dialog (see Figure 4 above).

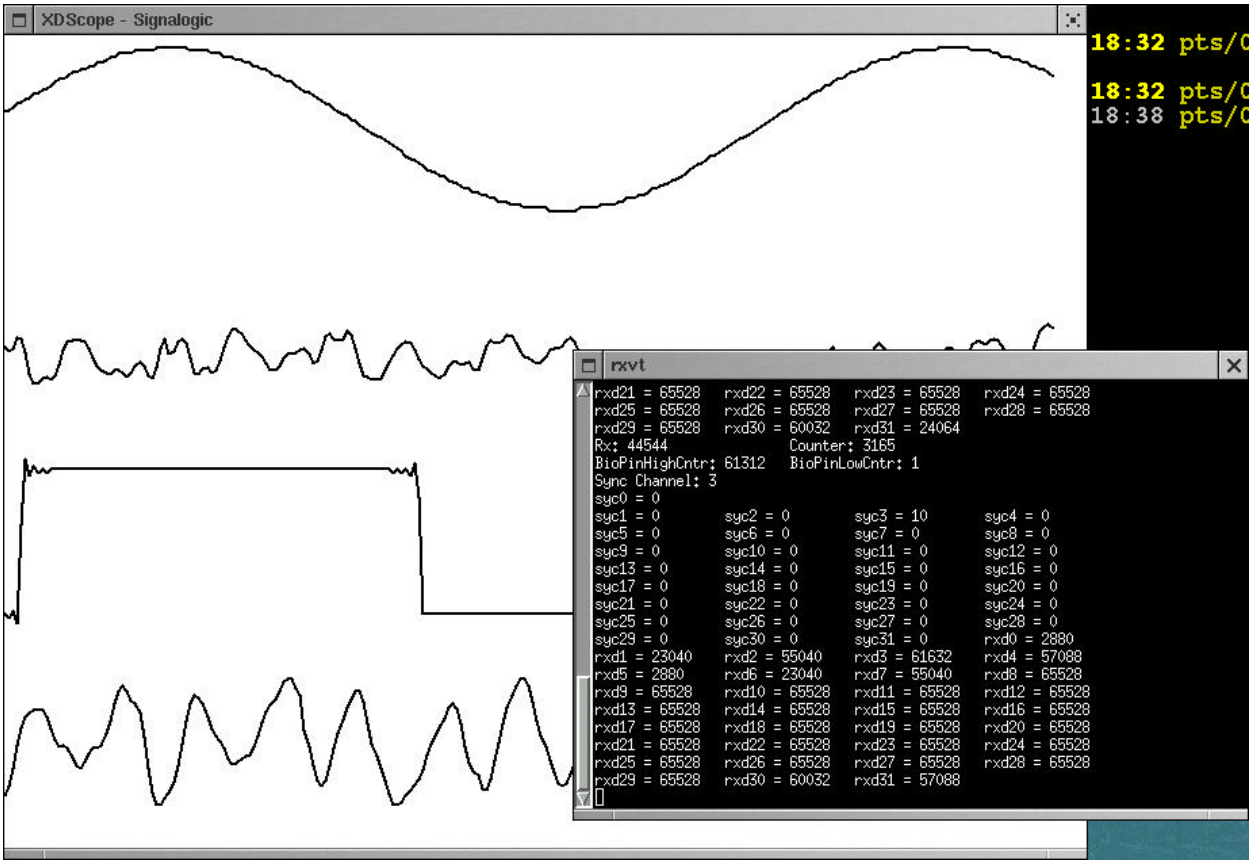

Figure 6 XDScope display with debug information shown

Below is another screen capture of the Hardware Manager dialog, showing some of the XTerm and other software displays that precede running the e1test program.

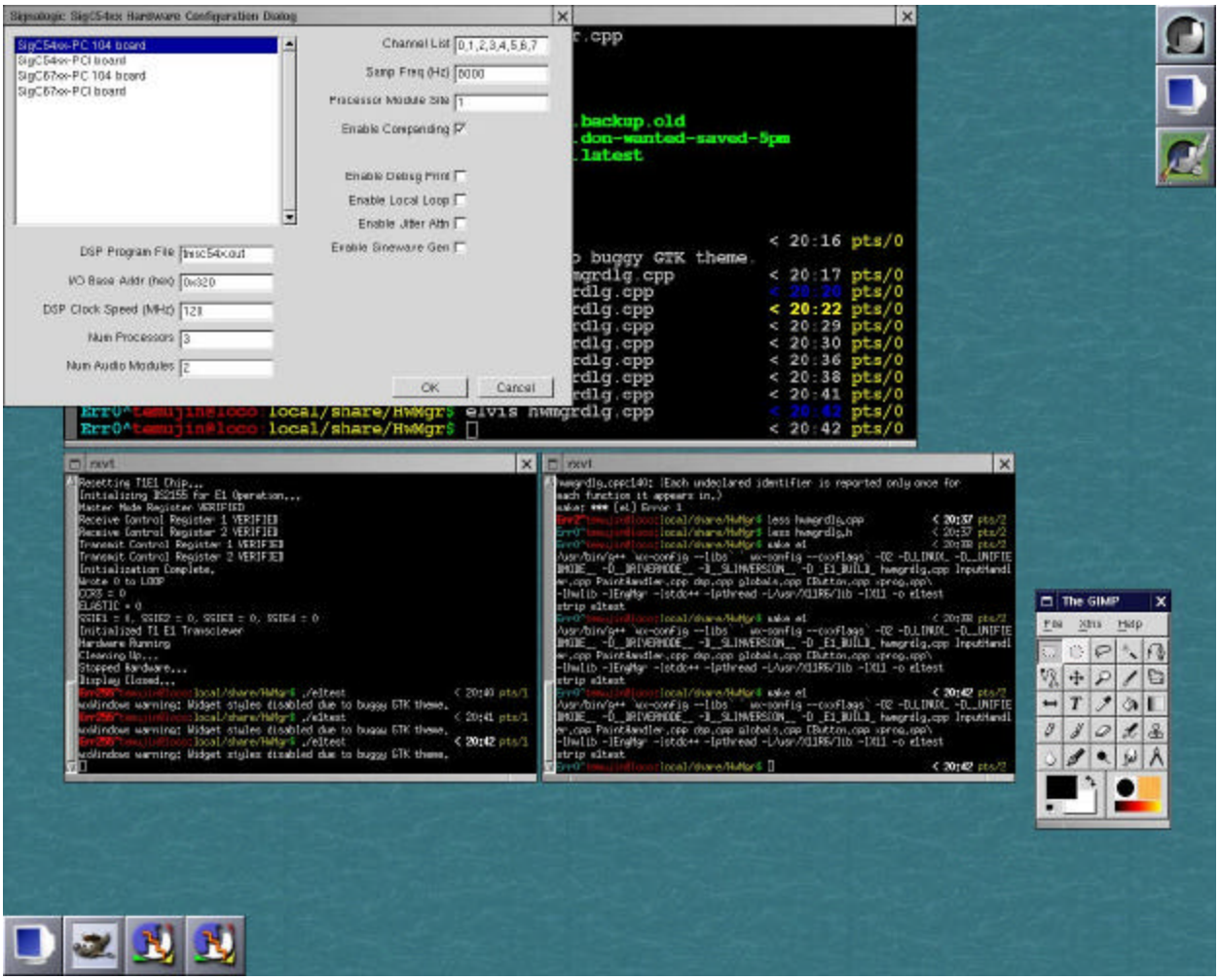

Figure 7 Hardware Manager dialog, with preceding background displays

2.2 Running the DUART Demo Program

***NOTE**: The following instructions assume that the SigFM.o module has been built from the SigFModule directory. If not, see build instructions in section 3.3.*

After initial system reboot, the kernel module must be loaded. See instructions in section 1.6 above, Linux Driver Install and Run Instructions.

Before running the DUART demo program, Hyperterminal must be set up on the PC that the RS-232 adapter is connected to.

To set up Hyperterminal, first run Start->Programs->Accessories->Communications->Hyperterminal.

Type in any name for the session and click OK.

Select the COM port the serial cable is attached to and click OK.

Select 19200 bit rate, 8 data bits, no parity, 1 stop bit, and Xon/Xoff Flow Control. Click OK.

To run the DUART demo program, first enter:

```
shell/> /sbin/insmod SigFM
```

This will install the SigFM.o module into the kernel so that the Signalogic software can run.

Then move to the subdirectory from which the DUART demo program will be run, as follows:

```
shell/> cd /usr/local/share/Comm-Prog
```

To run the Signalogic DUART demo program, enter:

```
shell/Comm-Prog> ./CommProg
```

A prompt should appear asking whether to run the program in normal mode or loopback mode. Type 1 for normal mode to generate character output internally or 2 for loopback mode to generate character output from character input.

This will run the DUART test. If DUART runs successfully, the screen will appear as shown in Figure 8 below:

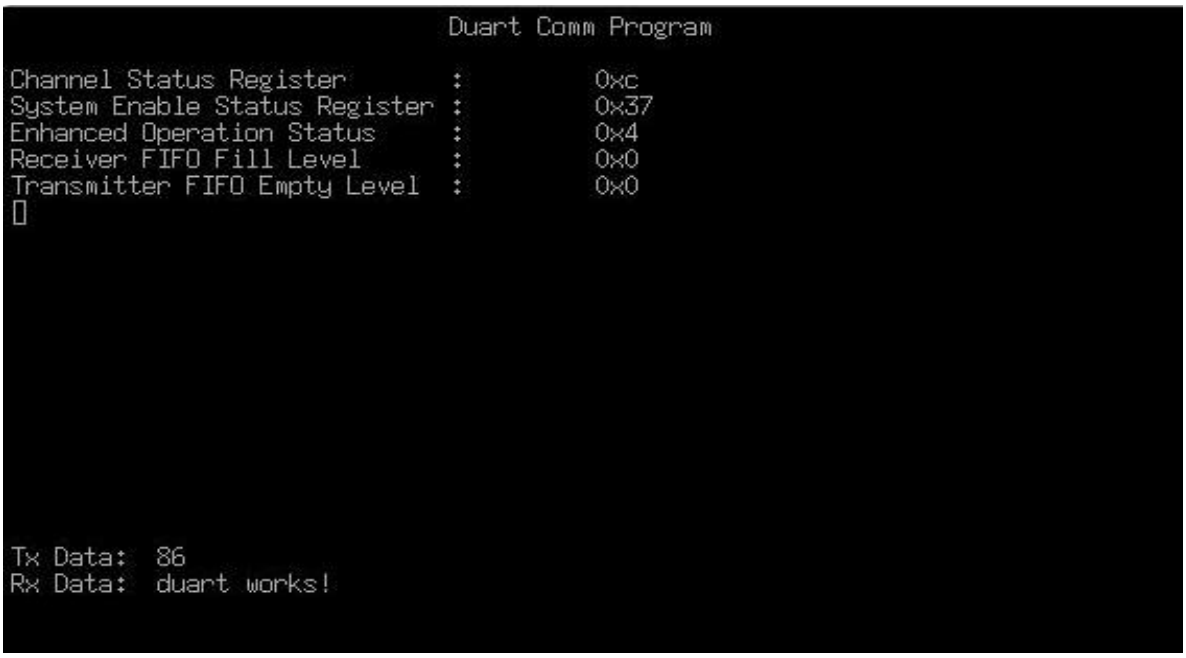
Duart Comm Program
Channel Status Register : 0xc
System Enable Status Register : 0x37
Enhanced Operation Status : 0x4
Receiver FIFO Fill Level : 0x0
Transmitter FIFO Empty Level : 0x0
Tx Data: 86
Rx Data: duart works!

Figure 8 Running DUART Comm Program

Note the values of the control or status registers should be 0xc, 0x37, 0x4, 0x0, and 0x0 respectively. For mode 1, *Tx Data:* should display numbers in a cycle from about 70 to 90. This means the DUART is transmitting capital letters from A to Z to Hyperterminal in a continuous cycle as shown below in Figure 9. When a key is pressed on the PC, the character should appear in Hyperterminal, and Rx Data should display the character pressed.

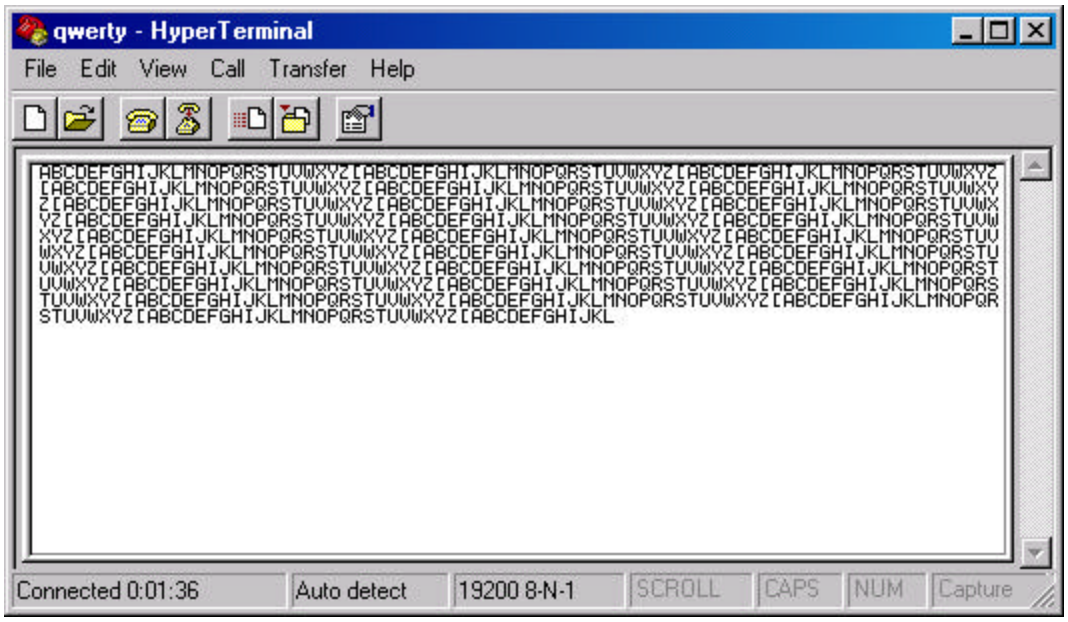

Figure 9 Hyperterminal on target PC

3 Rebuilding Software

To build the software modules listed in the subsections below, you will need to have root access. There are three (3) options for gaining root access:

1. Log-in as root user:

```
Login: root
password: veyung
```

2. Acquire sudo access rights to commands ldconfig, depmod, rmmod, and insmod. To gain sudo access rights, type:

```
sudo  <command>
```

followed by the command. Then enter your user password.

3. Switch user to root by typing

```
su
```

at the command prompt and enter the root password.

Notes

1) Under /usr/local/share/, the subdirectories HwMgr, EngMgr, SigFModule have attributes user nobody and group nobody.

2) On low-power consumption systems such as a Lippert CoolRunner II, a full make of the e1 build may take up to 5 minutes because the processor speed is relatively slow, about 300 MHz.

3) The methods above discuss root access. For information purposes, note that on the Lippert system, a normal user account has also been created with username temujin and password veyung.

3.1 Rebuilding the E1-Audio Demo Program

To build the Signalogic E1-Audio demo program from the main extraction folder (see section 1.5 above, Directory Structure), change to the HwMgr folder and execute the make command, as shown below:

```
shell/> cd HwMgr

shell/HwMgr> make e1
```

This will create a new e1test.o executable module, which is the E1-Audio version of the XDScope demo program that comes with the Linux version of DirectDSP software. To run the E1-Audio demo program, see section 2.1 above, Running the E1-Audio Demo Program.

Note that it is not necessary to first gain root access to rebuild e1test, since it is an application-level program.

3.2 Rebuilding the DUART Demo Program

To build the Signalogic DUART demo program from the main extraction folder (see section 1.5 above, Directory Structure), change to the Comm-Prog folder and execute the make command, as shown below:

```
shell/> cd Comm-Prog

shell/HwMgr> make CommProg
```

This will create a new CommProg.o executable module. To run the CommProg demo program, see section 2.2 above, Running the DUART Demo Program. Note that it is not necessary to first gain root access to rebuild CommProg, since it is an application-level program.

3.3 Rebuilding the LKM Driver

This section describes how to rebuild the LKM (loadable kernel module) driver and Hardware Library shared object module used by SigC54xx and SigC67xx boards under Linux.

Note that the LKM and Hardware Library shared object module should always be rebuilt together, to avoid any incompatibilities or mismatches that would affect the interface between these two software components.

First, log-in as root user (see section 1.8 above, Rebuilding Software). Then remove the driver from memory, by entering:

```
shell/> /sbin/rmmod SigFM
```

Next, from the main extraction folder (see section 1.5 above, Directory Structure), change to the SigFModule folder and execute the make command:

```
shell/> cd SigFModule

shell/SigFModule> make all
```

Last, re-install the LKM driver, as shown below

```
shell/SigFModule> insmod SigFM.o
```

NOTE: *In order for the "make all" command to work correctly, do not attempt to install the loadable kernel module driver or run individual pieces of the makefile separately without knowing EXACTLY what you are doing.*

3.4 Rebuilding Engine Manager

To rebuild the Engine Manager shared object module, follow the steps below.

First, log-in as root user (see section 1.8 above, Rebuilding Software). Then from the main extraction folder (see section 1.5 above, Directory Structure), change to the EngMgr folder:

```
shell/> cd EngMgr
```

Make and install the Engine Manager module:

```
shell/EngMgr> make all
```

3.5 Rebuilding Hardware Library

To rebuild the Hardware Library shared object module, follow the steps below.

From the main extraction folder (see section 1.5 above, Directory Structure), change to the SigFModule folder:

```
shell/> cd SigFModule
```

Make and install the Hardware Library module:

```
Shell/HwLib/> make all
```